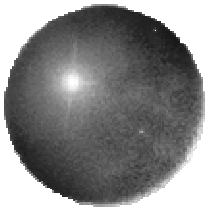
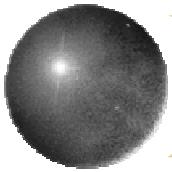


Deploying and Operating the SAM-Grid: lesson learned

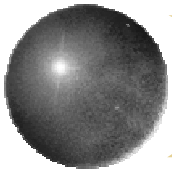
Gabriele Garzoglio for the SAM-Grid Team

Sep 28, 2004



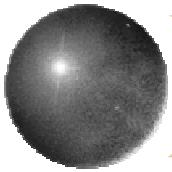
Overview

- Introduction to the SAM-Grid
- The SAM-Grid deployment and operations
- Lesson learned
  - Cluster
  - Grid/Fabric interface
  - Grid services



The SAM-Grid Project

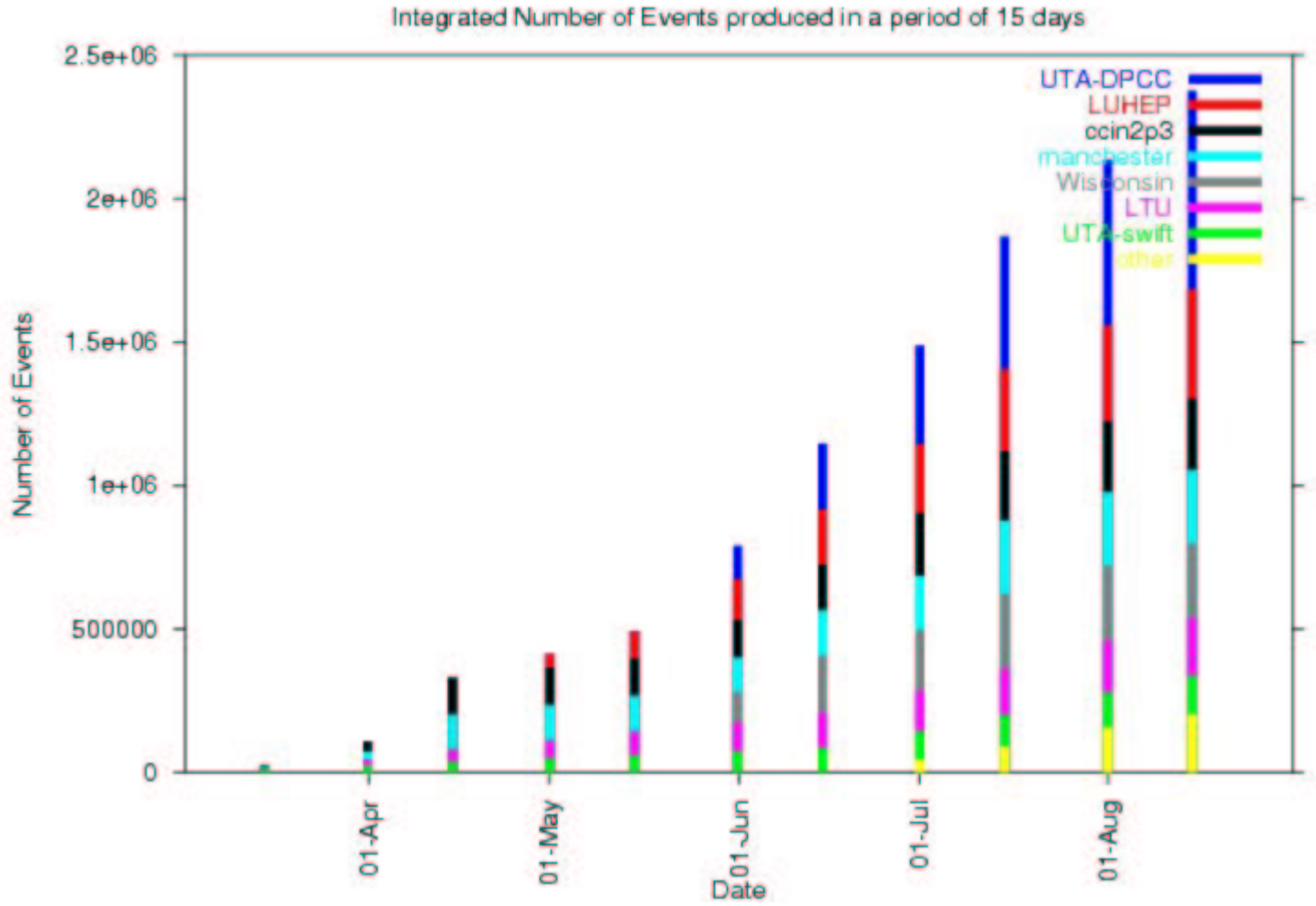
- **Mission**: enable fully distributed computing for DZero and CDF
- **Strategy**: enhance the distributed data handling system of the experiments (SAM), incorporating standard Grid tools and protocols, and developing new solutions for Grid computing (JIM)
- **History**: SAM from 1997, JIM from end of 2001
- **Funds**: received some funding from the Particle Physics Data Grid (US) and GridPP (UK)
- **People**: Computer scientists and Physicists from Fermilab and the collaborating Institutions

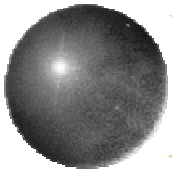


What is SAM-Grid used for?

- Montecarlo production for DZero
  - From March 2004 produced > 2,000,000 events, equivalent to 11 yrs GHz computation
- Other Activities:
  - Extending the infrastructure to enable data reconstruction for DZero
  - Montecarlo production for CDF at the prototypical stage

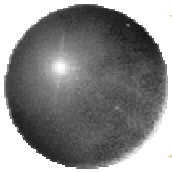
Montecarlo Production Events





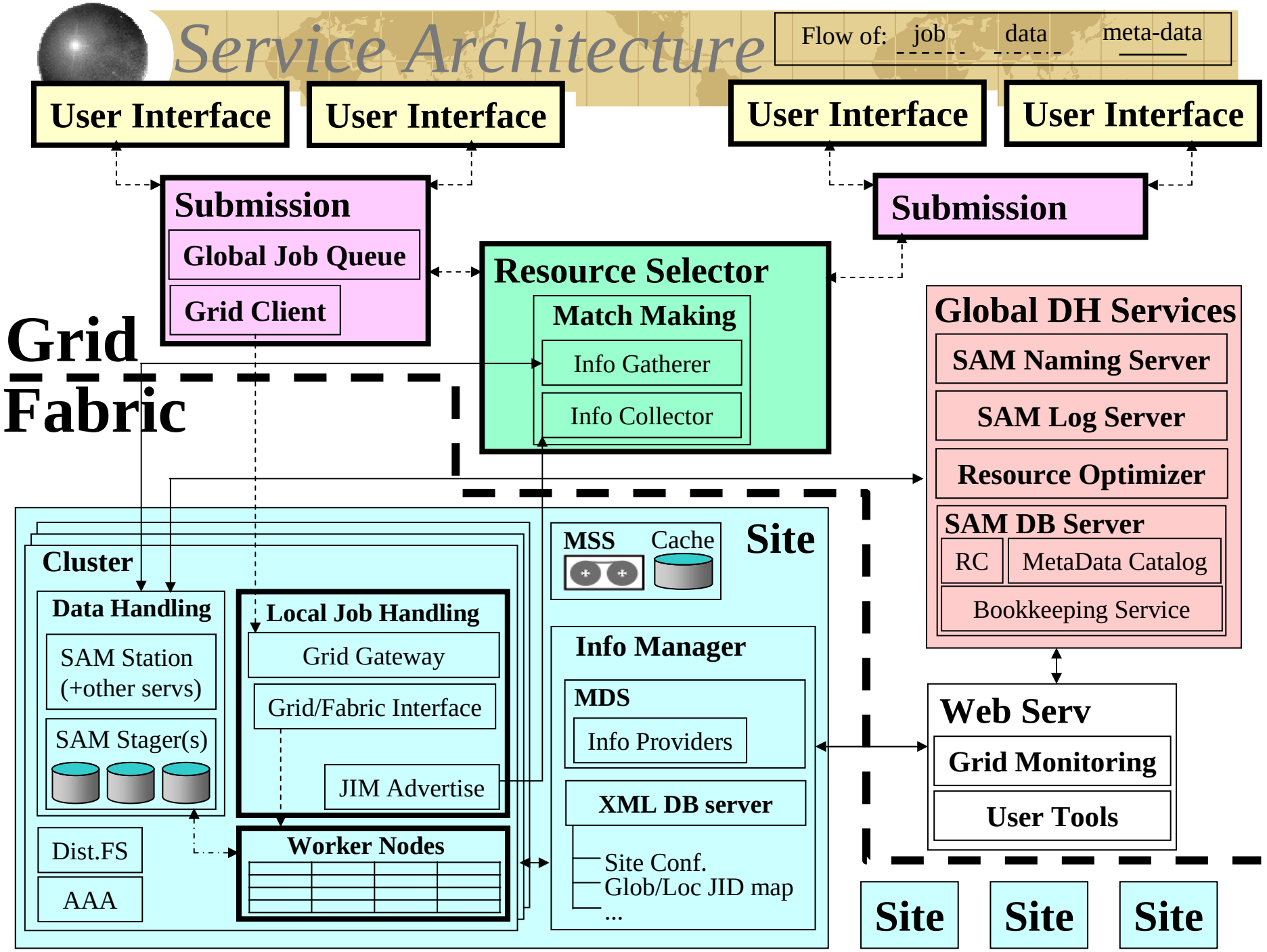
Overview

- ✓ Introduction to the SAM-Grid
- ➢ The SAM-Grid deployment and operations
- Lesson learned
  - Cluster
  - Grid/Fabric interface
  - Grid services



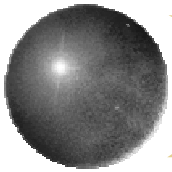
The Deployment Phase

- The initial deployment took 3 months: Jan - Mar 2004
- The *inefficiency* in event production due to the grid infrastructure improved from 40% to 1-5%
  - Inefficiency of the infrastructure = 1 - (events produced / events requested)
- This talk focuses on the main sources inefficiencies and how we mitigated them


Service Architecture
Flow of: job data meta-data
User Interface
User Interface
User Interface
User Interface
Submission
Global Job Queue
Grid Client
Submission
Resource Selector
Match Making
Info Gatherer
Info Collector
Grid Fabric
Global DH Services
SAM Naming Server
SAM Log Server
Resource Optimizer
SAM DB Server
RC
MetaData Catalog
Bookkeeping Service
Site
Cluster
MSS
Cache
Data Handling
SAM Station (+other servs)
SAM Stager(s)
Local Job Handling
Grid Gateway
Grid/Fabric Interface
JIM Advertise
Info Manager
MDS
Info Providers
XML DB server
Site Conf.
Glob/Loc JID map
...
Web Serv
Grid Monitoring
User Tools
Dist.FS
AAA
Worker Nodes
Site
Site
Site

The Deployment Model

- Every site provides a gateway node where experts + local contacts can install the SAM-Grid software:
  - Standard middleware (VDT), Grid/Fabric interface, VO Services client code
- VO-specific services run at the site:
  - SAM, JIM Monitoring, Local Scheduler, Local Storage
- No software/daemon required at the worker nodes of the cluster

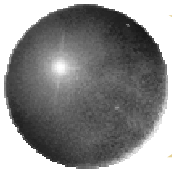


Status of the Deployment

- A dozen institutions currently part of the grid
- ~50% stable enough to be used for production
- US Institutions
  - FNAL, UW Madison, UTA, LUHEP, LTU, OSCER, OUHEP
- Non-US Institutions
  - IN2P3 (Fr), Oxford (UK), Manchester (UK), Prague (Cz), GridKa (De), Sprace (Br)

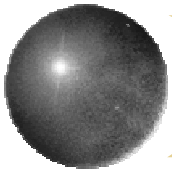
The Operation/Support Model

- A few production users can submit from their laptop to any SAM-Grid site
- The software at each site is *uniform* and *adapts* to the local fabric configuration
- The JIM infrastructure is currently maintained by 1 FTE + local contacts.
  - This improves the previous model, where an expert per site was necessary to maintain the specific local production mechanisms



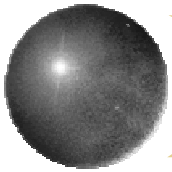
Overview

- ✓ Introduction to the SAM-Grid
- ✓ The SAM-Grid deployment and operations
- Lesson learned
  - Cluster
  - Grid/Fabric interface
  - Grid services



System Configuration Problems 1

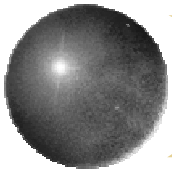
1. Time synchronization of the worker nodes
   - The Grid Security Infrastructure relies on the machine clock to determine the validity of the security tokens
   - Administrators: please run ntpd !
   - We also introduced artificial delays at the worker nodes to avoid "*Proxy not yet valid*" errors



System Configuration Problems 2

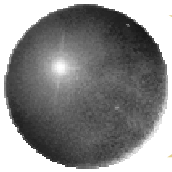
2. The Black Hole effect
   - Even if a single node in the cluster is mis-configured and makes its jobs crash, the batch system keeps sending idle jobs to it: the whole queue of jobs will crash.
3. The Batch System does not immediately show up the jobs submitted to it or it times out
   - When the Grid asks the status of the jobs and cannot find them, it thinks that they are finished: resource leak!

- Both problems have been solved writing an "idealizer" (level of abstraction) in front of the batch system. In this code we can exclude statistically bad nodes, retry polling commands, etc.



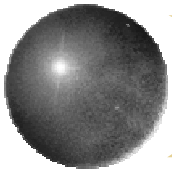
System Configuration Problems 3

4. The worker nodes do not know their domain name
   - Our infrastructure wants to know… is this really SAM-Grid specific?
5. Running gridftp transfers between worker and head node *within a private network* is tricky
   - Gridftp works in active mode only: the server at the head node may not be able to open the port to the client at the worker node
   - Solution: give the head node a private network interface



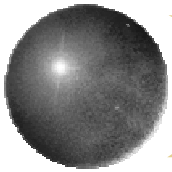
System Configuration Problems 4

6. Plan the OS upgrades with the system administrators… or be resilient to it
   - "We upgraded the worker nodes to RH9 and forgot to tell you…"
7. Negotiate/Study the policy limits
   - Jobs have been killed or slowed down by batch system CPU limits, data handling file transfers limits, probability of job preemption ~1, …



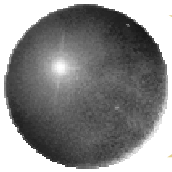
Overview

- ✓ Introduction to the SAM-Grid
- ✓ The SAM-Grid deployment and operations
- Lesson learned
  - ✓ Cluster
  - ➢ Grid/Fabric interface
  - Grid services



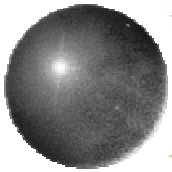
Gateway and VO Problems

- Most of our work went in the interface between the Grid and the Fabric
- The standard Globus job-managers are not sufficiently…
  - …flexible: they expect a "standard" batch system configuration. None of our sites was that standard.
  - …scalable: a process per grid job is started up at the gateway machine. We want/need aggregation.
  - …comprehensive: they interface to the batch system only. How about data handling, local monitoring, databases, etc.
  - …robust: if the batch system forgets about the jobs, they cannot react. We have written the "idealizers" for this.
- To address these issues we had to write a "thick" Grid/Fabric interface (jim-job-manager). The drawback of this approach is that it complicates the local configuration.



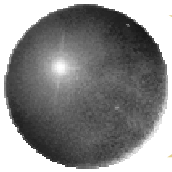
Overview

- ✓ Introduction to the SAM-Grid
- ✓ The SAM-Grid deployment and operations
- Lesson learned
  - ✓ Cluster
  - ✓ Grid/Fabric interface
  - ➢ Grid services



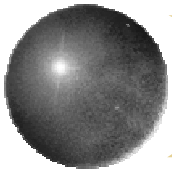
Grid Services Problems 1

- Scalability of the semi-central services
  - access to the central data handling database is organized in a 3-tiers architecture
  - the middle tier couldn't cope with 200 jobs starting up at the same time, asking for data
  - we had to introduce retrials with exponential back off to mitigate the problem. We also aggregate access from the gateway node for the information that is common to all processes.



Grid Services Problems 2

- Firewalls: understand the network topology of your grid
  - System administrators generally are willing to open ports to a certain list of nodes when the software is installed
  - Maintaining the configuration up to date as new installation are deployed is difficult
  - For core services, such as data movement, the SAM-Grid can route data via delegation if direct transfers are not possible



Conclusions

- The SAM-Grid is an integrated grid system for job, data and information management for HEP
- It is used in "production" for DZero montecarlo since March 2004.
  - We are working on data reconstruction for DZero and montecarlo generation for CDF
- During deployment and operations we had to overcome problems at the level of
  - the systems: careful administration is crucial
  - the Grid/Fabric interface: we need a "thick" interface
  - the Grid services: be careful about scalability and network topology